Classificazione dataset FoodX-251

Elia Gaviraghi 869493
Erba Sandro 856327

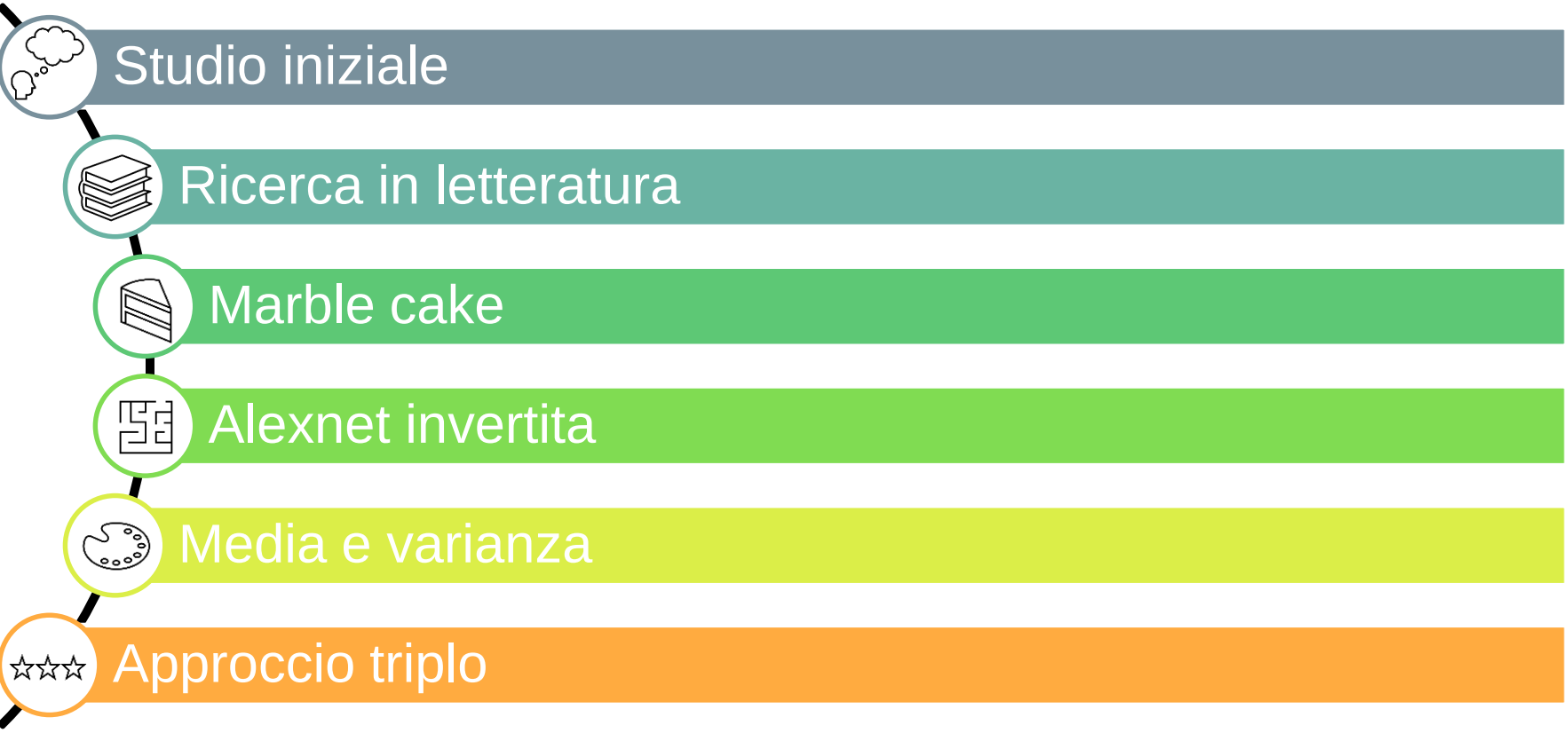
UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

Visual Information processing and management – 2023/24

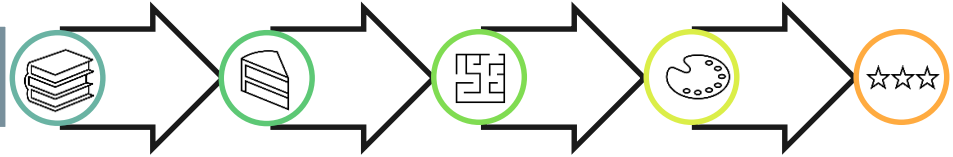
Indice

- Pulizia del training set
- Addestramento rete e classificazione del test set
- Pulizia e classificazione del degraded test set
- Appendice

Pulizia del training set



Studio iniziale



Assunzioni:

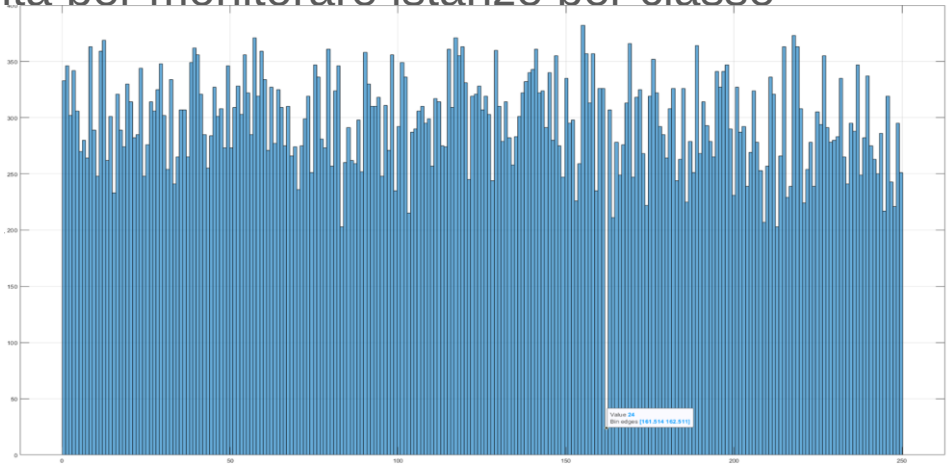
- Le immagini sporche sono equamente distribuite in ogni classe
- Il test set contiene unicamente immagini pulite e correttamente etichettate
- ?

Ipotesi:

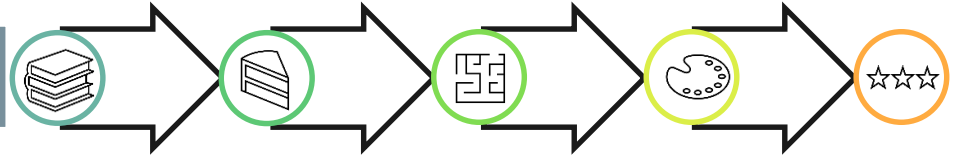
- Clustering con distanza centroidi
- Rete neurale addestrata sul test
- Approccio tramite descrittori dell’immagine

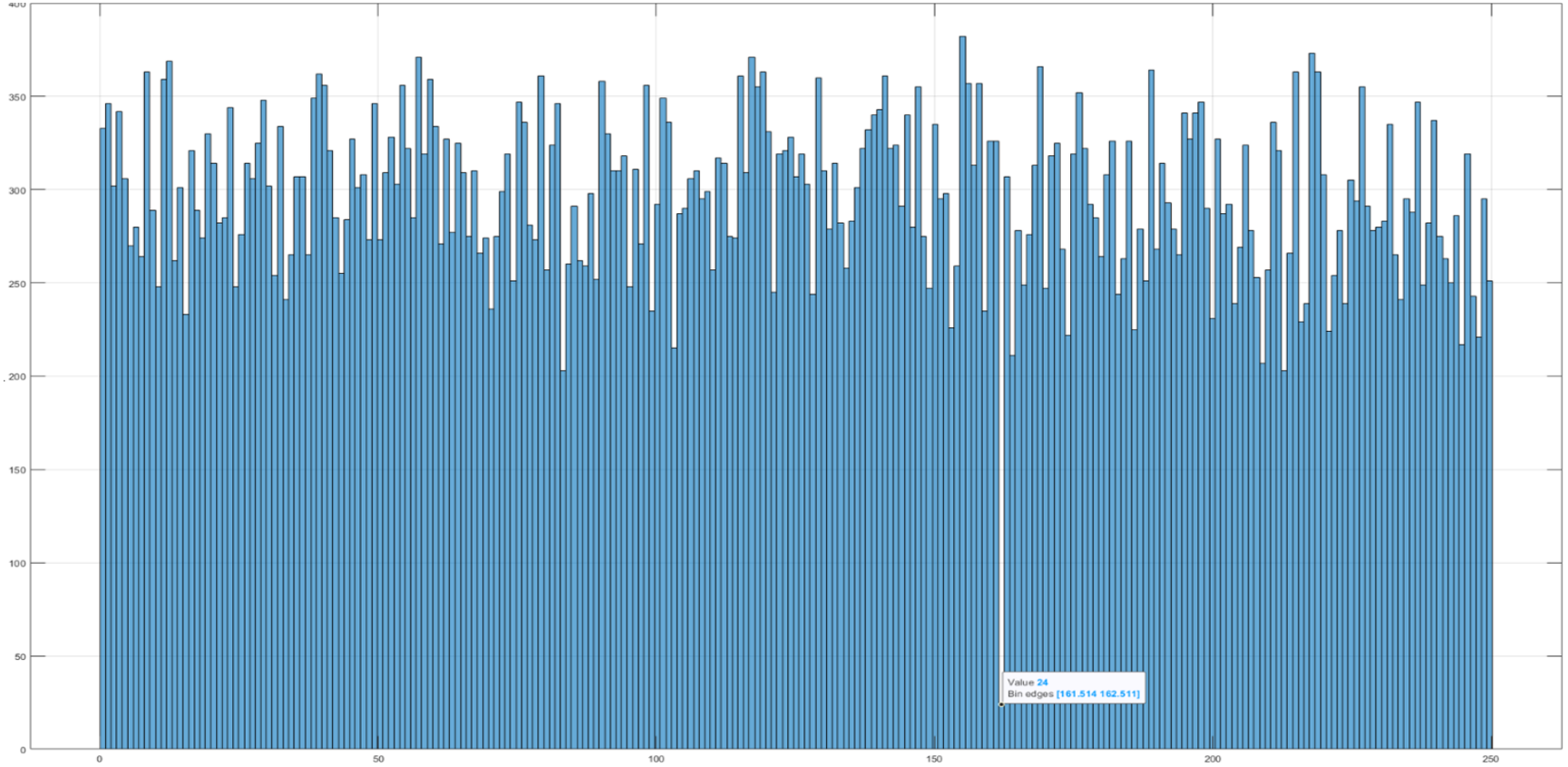
Studio iniziale

- Analisi delle classi e delle immagini presenti
- Ci sono immagini non di cibo così come immagini di cibo ma con label scorrette
- Divisione in 251 folder del train set
- Grafici di numerosità per monitorare istanze per classe

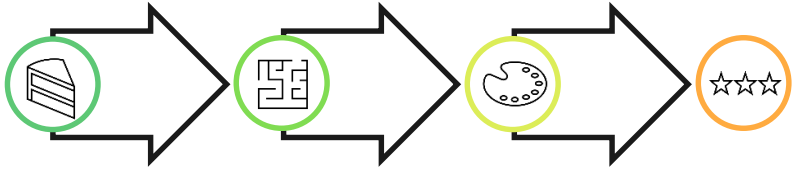


Studio iniziale





Ricerca in letteratura



- *FoodX-251: A Dataset for Fine-grained Food Classification*: Introduce il dataset anticipando che «piatti popolari sono classi con un gran numero di campioni» e viceversa. Usa ResNet-101 con ADAM, lr = 5e−5 schedulato di un fattore di 10 ogni 10 epochs.
- *An Artificial Intelligence-Based System to Assess Nutrient Intake for Hospitalised Patients*: Parla di malnutrizione per pazienti di ospedali. Usa una ResNet50 per initial feature maps, ma ha anche vari blocchi di conv, usa data aug. anche con le GAN
- Libreria FastDup per riconoscimento degli outliers.
- iFood - 2019 at FGVC6 challenge in cui si usa questo dataset.

Marble cake

Quasi tutte le classi sono bilanciate, tranne la 162 “marble cake”.
Questa ha solo 117 (non 24?) immagini e una trentina sono corrette.

- Soluzione possibile: merging con 119 “coffee cake” che è molto simile, ma non valida ai fini del progetto.
- Soluzione scelta: pulizia a mano e data augmentation.









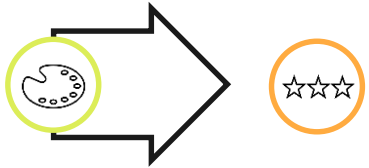
Marble cake

Riflessione

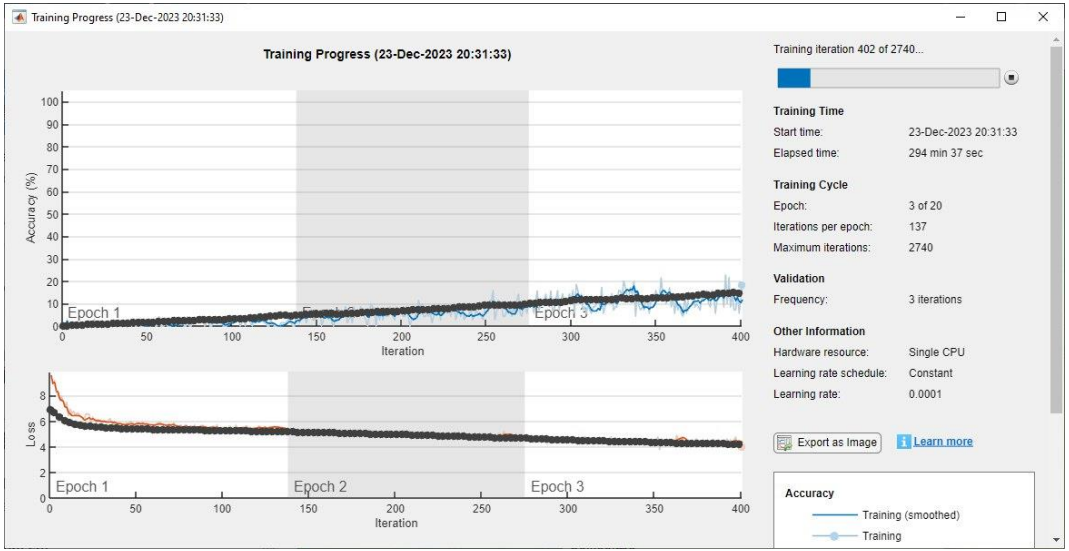
Rotazione

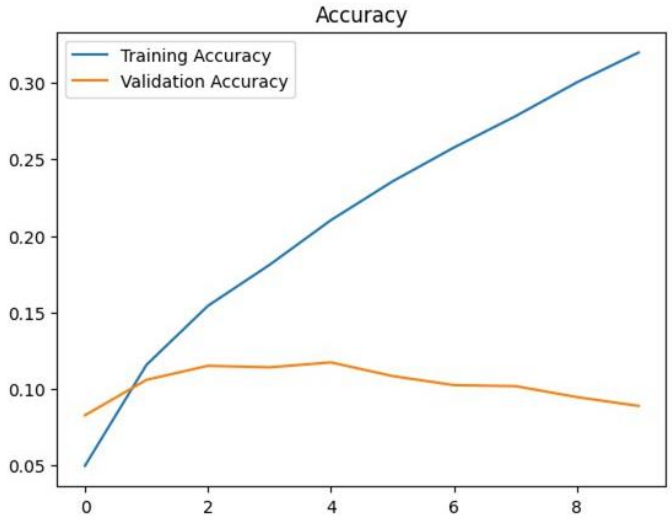
Traslazione

Alexnet invertita



Fine-tuning di una CNN pre-addestrata su imagenet usando il test set, in quanto questo è pulito. Enormi tempi di computazione per scarsi risultati. Si sarebbe potuto continuare in questa direzione, ma a discapito di troppo tempo.





Media e varianza

- Calcolare media e varianza per ogni classe;
- Eliminare il 20% di ogni classe;
- Rapido, ma richiede di settare le soglie a mano;
- Feature non troppo descrittive per tutto il training set.

Sicuramente NON un macaron

Potrebbero essere dei macaron

☆☆☆ Approccio triplo

Sulla base del precedente, per ogni classe estrarre le feature:

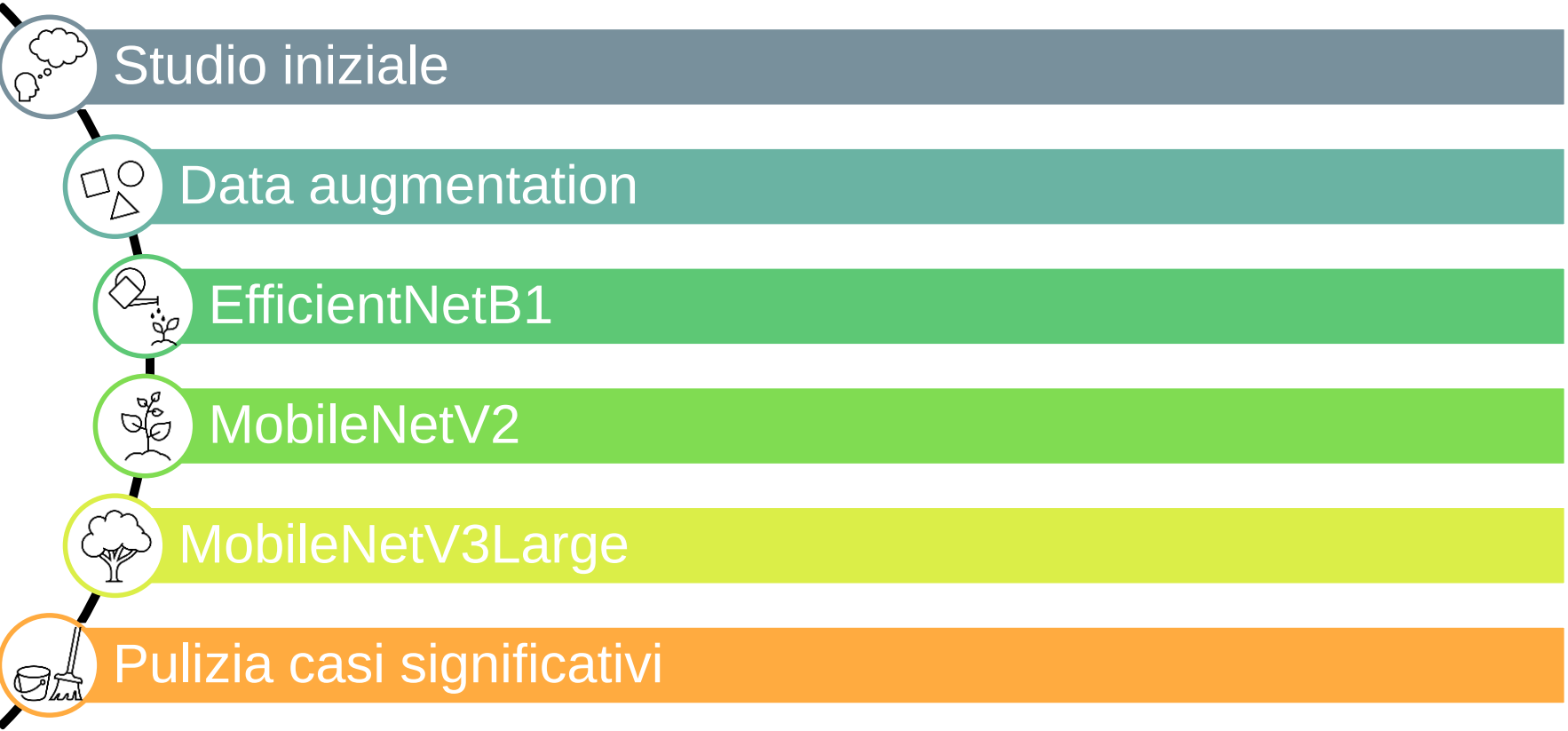
- CNN ?
- HOG ?
- texture con media e varianza

Fare clustering con queste e valutare gli outliers trovando la distanza dai centroidi.

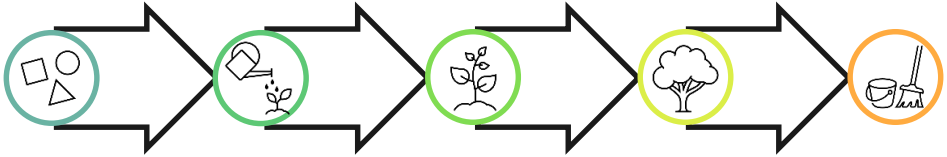
```
>> clustering
read dataset
Classe attuale 0
Elapsed time is 122.404676 seconds.
classe 0 con media 68 var 5100 outliers rimossi 12.20%
classe 0 con distanza centroide 0.50 outliers rimossi 20.04%
Outliers totali rimossi classe 0 è 28.05%
Classe attuale 1
```

?

Addestramento rete e classificazione del test set



Studio iniziale



Assunzioni:

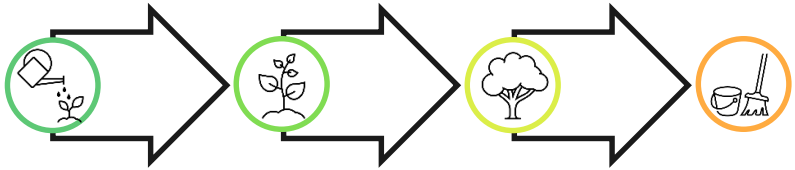
- Ogni immagine ha esattamente una label ?
- ?

Ipotesi:

- Addestramento rete da zero
- Rete addestrata su Imagenet con KNN ? finale
- Rete addestrata su Imagenet con FC finali
- Utilizzo di Colab per maggiore computazione

Documentazione Keras con elenco di models e come usarli.

Data augmentation



Originale

Rotazione

Flip orizzontale

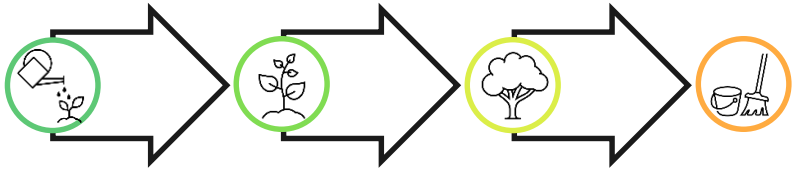
Prospettiva

Blur

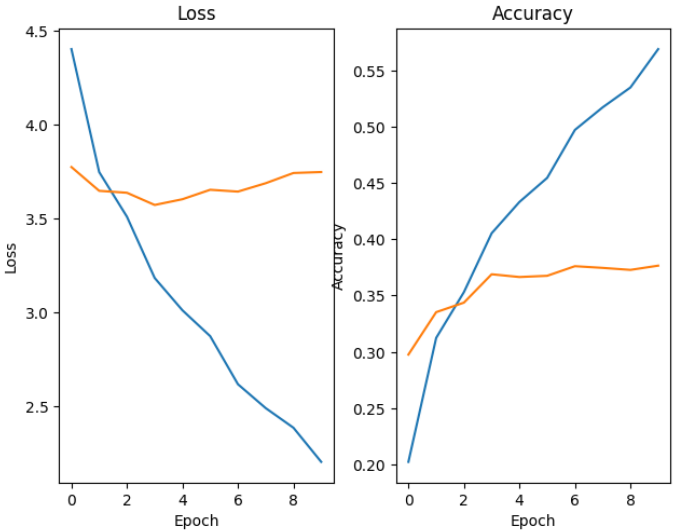
Noise gaussiano

Compressione

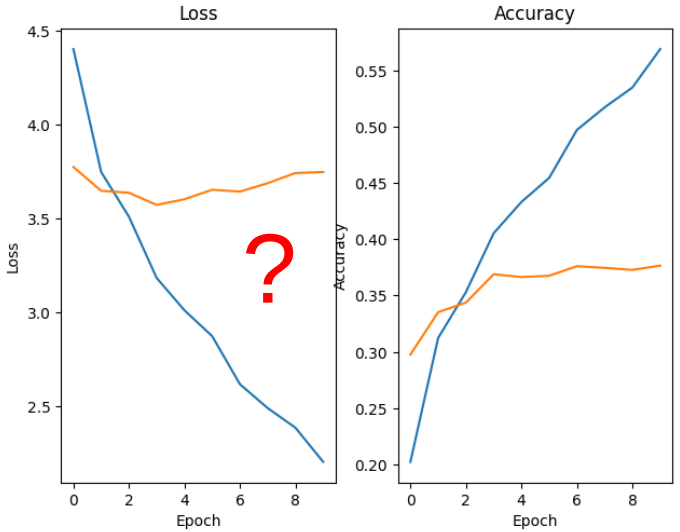
Data augmentation



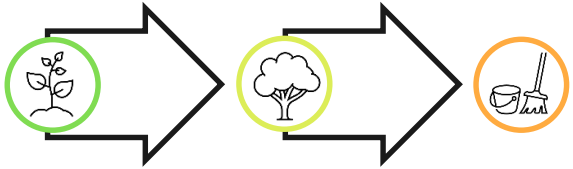
Train senza data aug.



Train con data aug.



EfficientNetB1



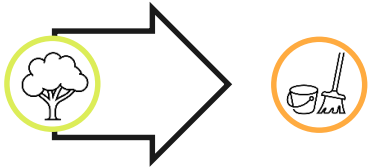
Buoni risultati, ma viene scartata perché non è possibile farci una successiva fase di train con i layer finali scongelati.

Scelte architetturali ?

Clean

Degraded

MobileNetV2



Regge lo sbloccamento di molti layer, tant’è che è possibile concatenare tre fasi di train, una successiva all’altra.

Scelte architetturali ?

Clean

Degraded

MobileNetV3Large

Migliore di quella precedente, sarebbe carino mettere una sorta di pipeline.

Scelte architetturali ?

Clean

Degraded

Pulizia casi significativi

(nelle richieste: analisi visuale di casi considerati significativi)
Tabellina con 10 peggiori classi in base al f1-score. Mostrare quelle poco numerose o altre cause.

Pulizia casi significativi

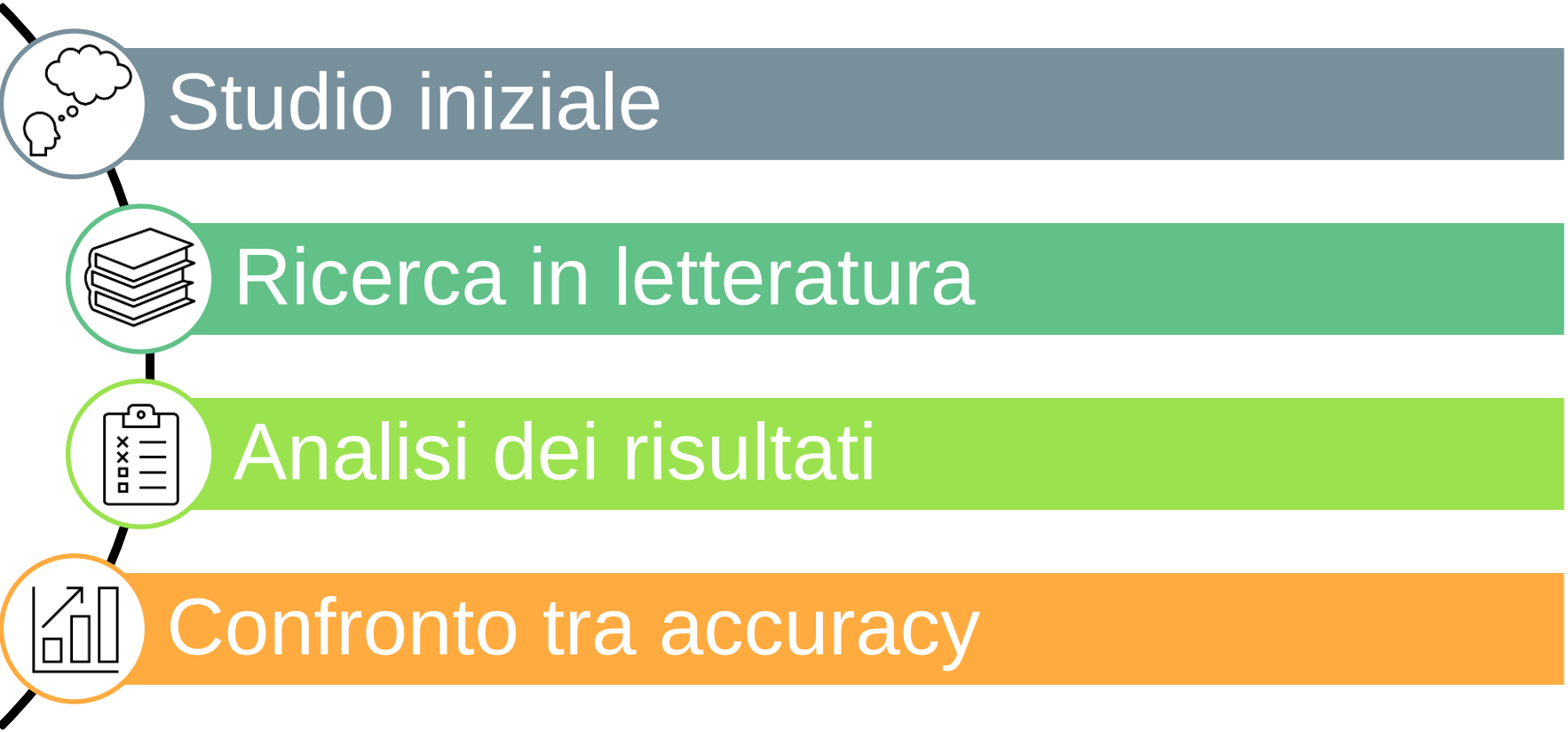
Modello finale con:

Dati del prof

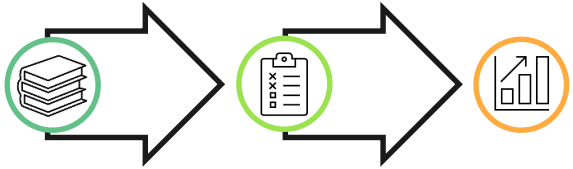
Dati post approccio triplo

Dati puliti a mano

Pulizia e classificazione del degraded test set



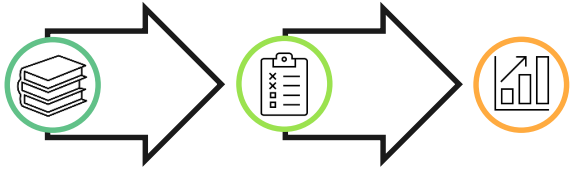
Studio iniziale



Assunzioni:

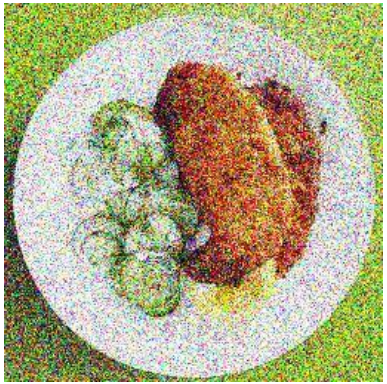
Ipotesi:

Studio iniziale



Rilevamento e classificazione dei tipi di degradazione presenti:

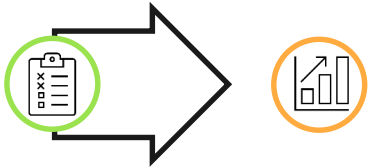
Noise gaussiano



Compressione JPEG

Blur

Ricerca in letteratura



Fast Noise Variance Estimation:

- Presenta un semplice metodo per stimare la varianza del rumore gaussiano additivo a media zero.
- Quasi insensibile alle strutture in un'immagine.
- Necessita di 14 operazioni per ogni pixel.
- Filtra l'immagine con la matrice N e la confronta con l'originale.

L1 =

0	1	0
1	-4	1
0	1	0

L2 =

$\frac{1}{2}$	0	$\frac{1}{2}$
0	-2	0
$\frac{1}{2}$	0	$\frac{1}{2}$

(c'è 1/2 ma non prende la frazione)

N = 2*(L1 – L2) =

1	-2	1
-2	4	-2
1	-2	1

Analisi dei risultati

- Media:
- Valore minimo:
- Valore massimo:

Threshold: 12?

Threshold: 27?





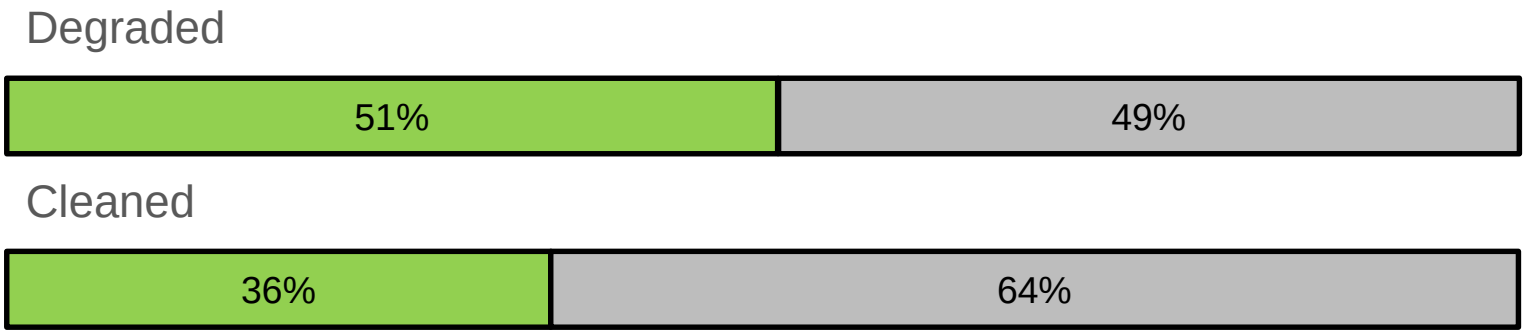

Percentile: 21% ?

Percentile: 66% ?

Percentile: 13% ?

Confronto tra accuracy

Si confronta l’accuracy di MobileNetV3Large sul degraded prima e dopo la pulizia



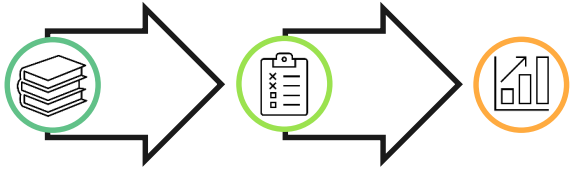
Category search

Studio iniziale

Approccio scelto

Analisi dei risultati

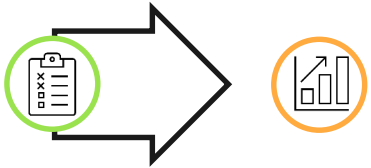
Studio iniziale



Assunzioni:

Ipotesi:

Approccio scelto

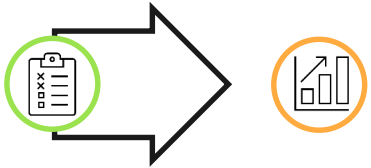


Parte che fai solo una volta e basta:

1. prendi il training set che abbiamo
2. prendi una MobileNetV3 (o qualsiasi altro modello) con i pesi di imagenet
3. per ogni immagine del training set, estrai le feature con la mobileNet e aggiungile a un mega array

Finito il tutto, salvati l'array

Approccio scelto



Parte che fai a ogni query:

1. immagine in input
2. estrai dall'immagine in input le feature con la stessa MobileNetV3, ottieni un array
3. confronta l'array appena ottenuto piccolo piccolo con tutto l'array gigante che ha tutte le feature di ogni immagine del training set
4. prendi le 10 immagini con feature meno distanti
5. fine

Analisi dei risultati

Appendice

- InceptionV3 (Sandro) 40%
- Più info sul modello finale, come è fatto
- Memino malandrino
- Possibili altre cose (oppure eliminare appendice)

Ricerca di significato di POI ha prodotto: